

Minería de datos y Patrones

Version 2024-I

Proyecto

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

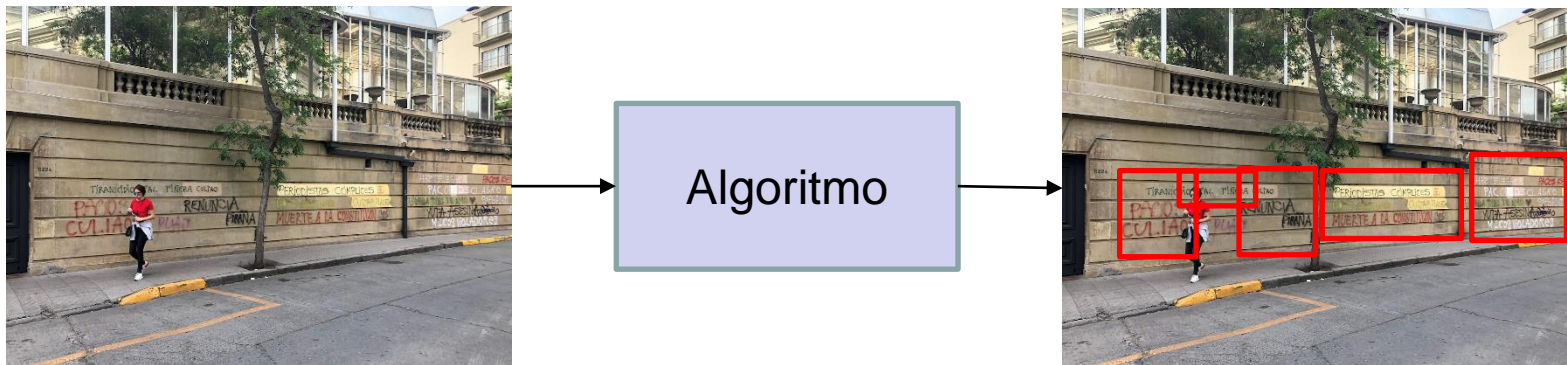
Ingeniería Electronica

Universidad Popular del Cesar

Proyecto del 2020

Detección de Rayados

Desarrollar un algoritmo que detecte las zonas de la imagen con rayados en las paredes.



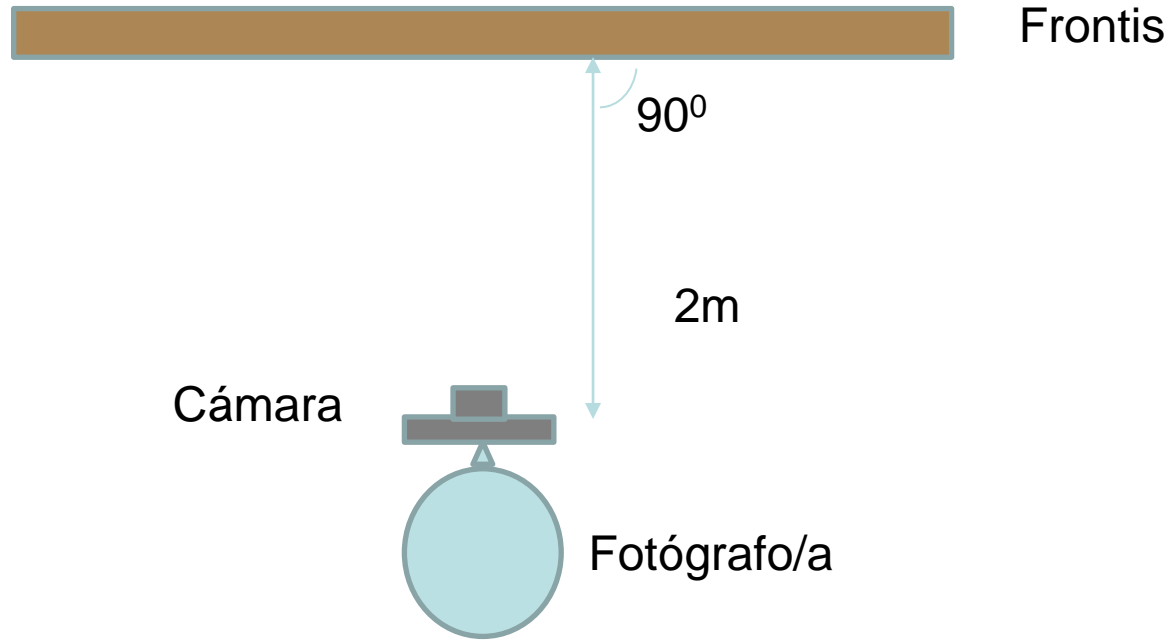
En este problema de reconocimiento de patrones hay dos clases:
Clase 0: No-Rayado y Clase 1: Rayado

Datos de Entrenamiento

Tomar fotos de la siguiente manera: pararse en la vereda a unos dos metros del frontis (pared / reja / portón / etc.) y tomar una foto que sea perpendicular al frontis.

Usar la luz del día, sin sombras, sin flash.

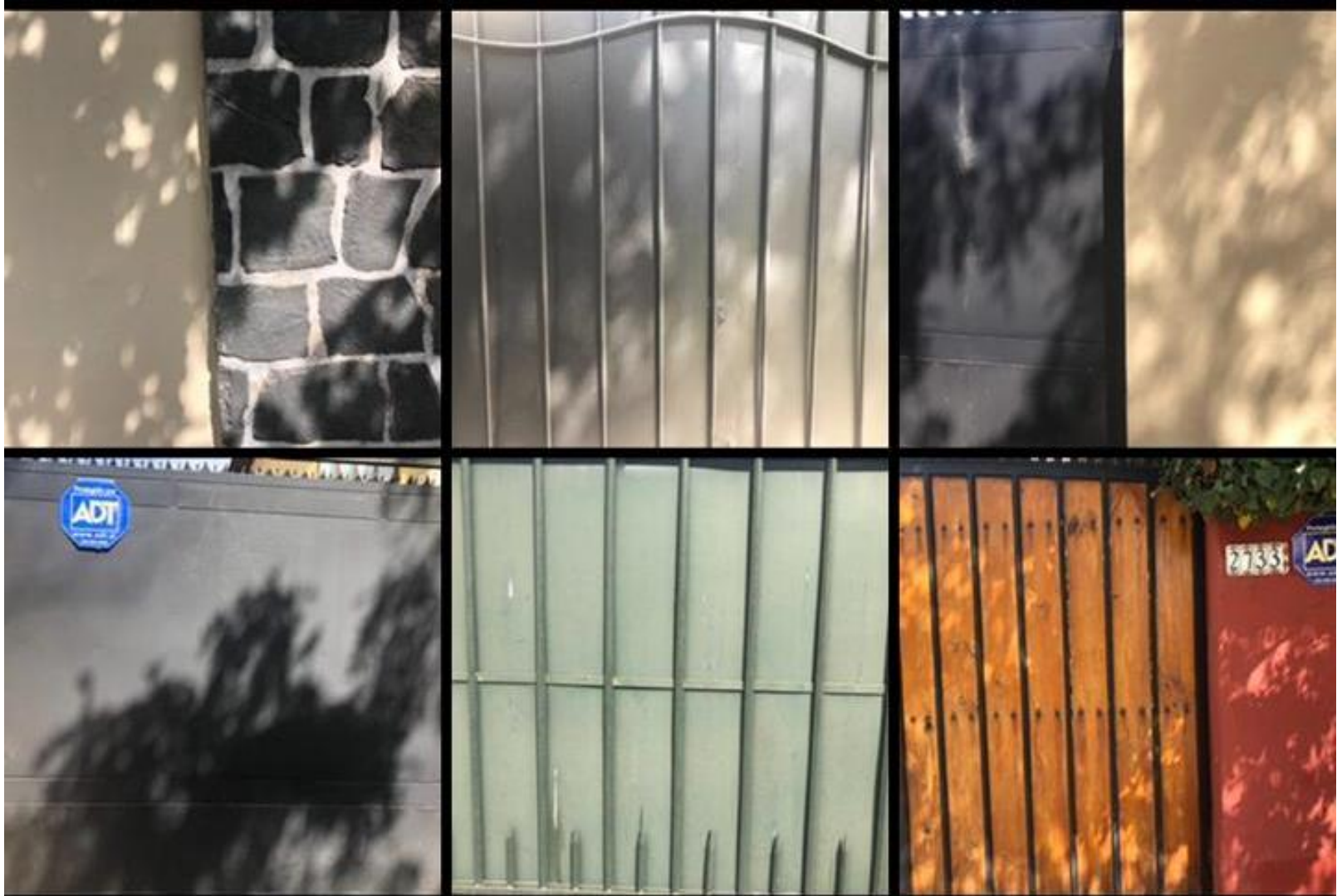
Top View



Datos de Entrenamiento (1/2)

La foto debe cubrir el 100% de una sola clase, es decir la foto debe contener un 100% rayado, o 100% no rayado.

Fotos Clase 0: No Rayadas



Fotos Clase 1: Rayadas



Datos de Entrenamiento (2/2)

Sólo una porción de la foto debe estar rayada. Adicional a la foto se debe entregar las coordenadas del/os bounding box/es que cubran el rayado.

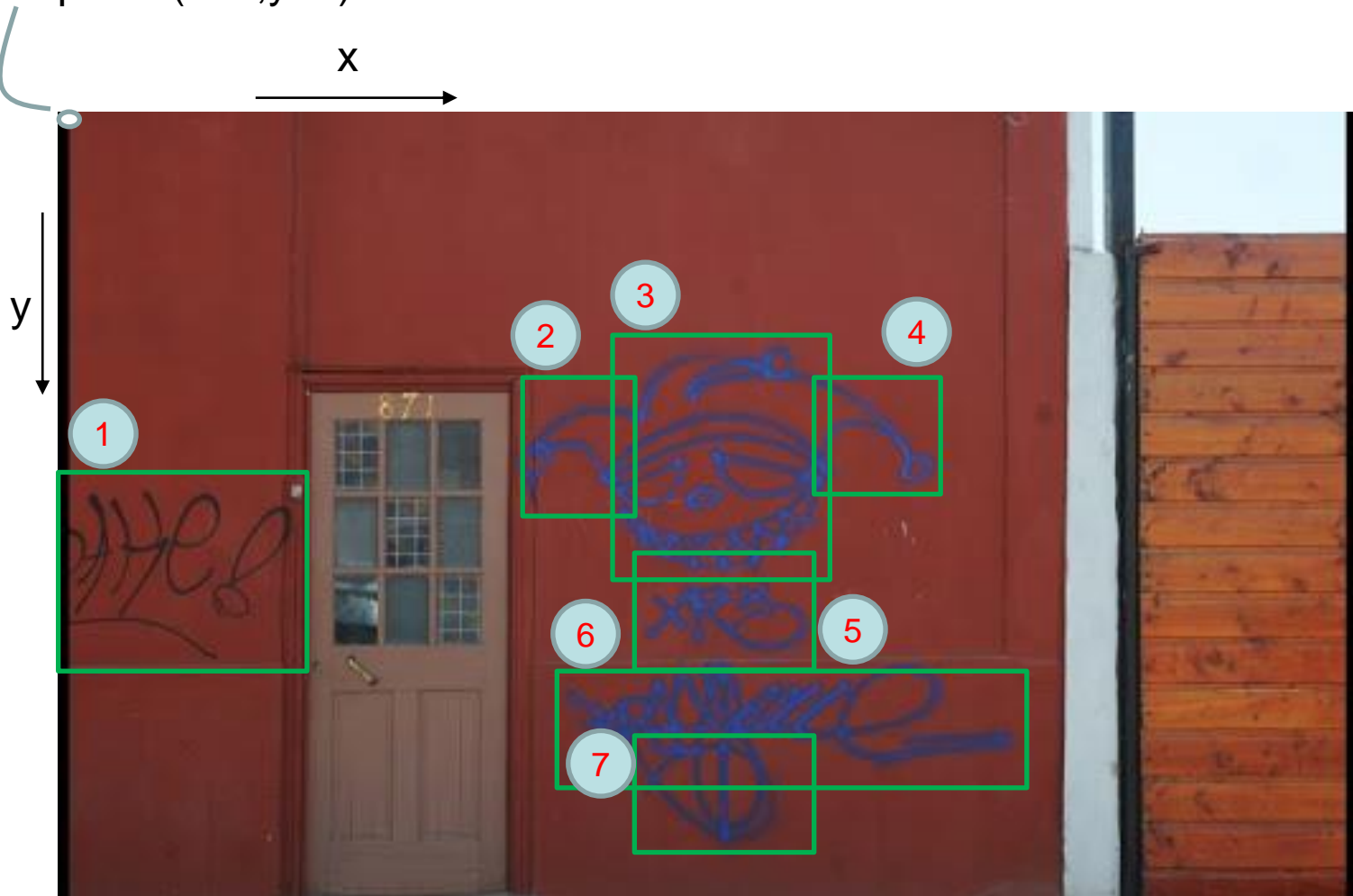
Datos de Entrenamiento (2/2)



(imagen original)

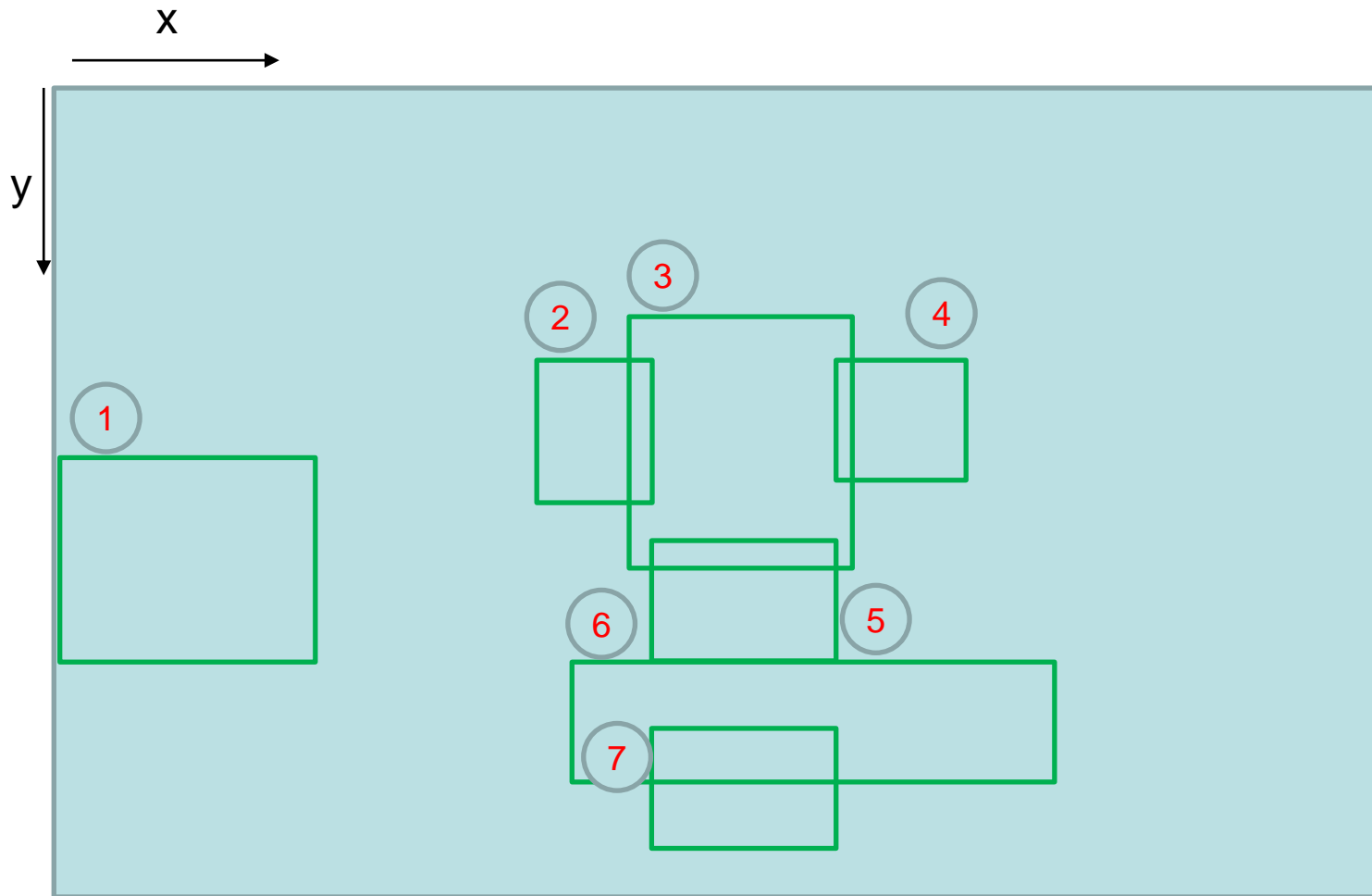
Datos de Entrenamiento (2/2)

Primer pixel: (x=1,y=1)



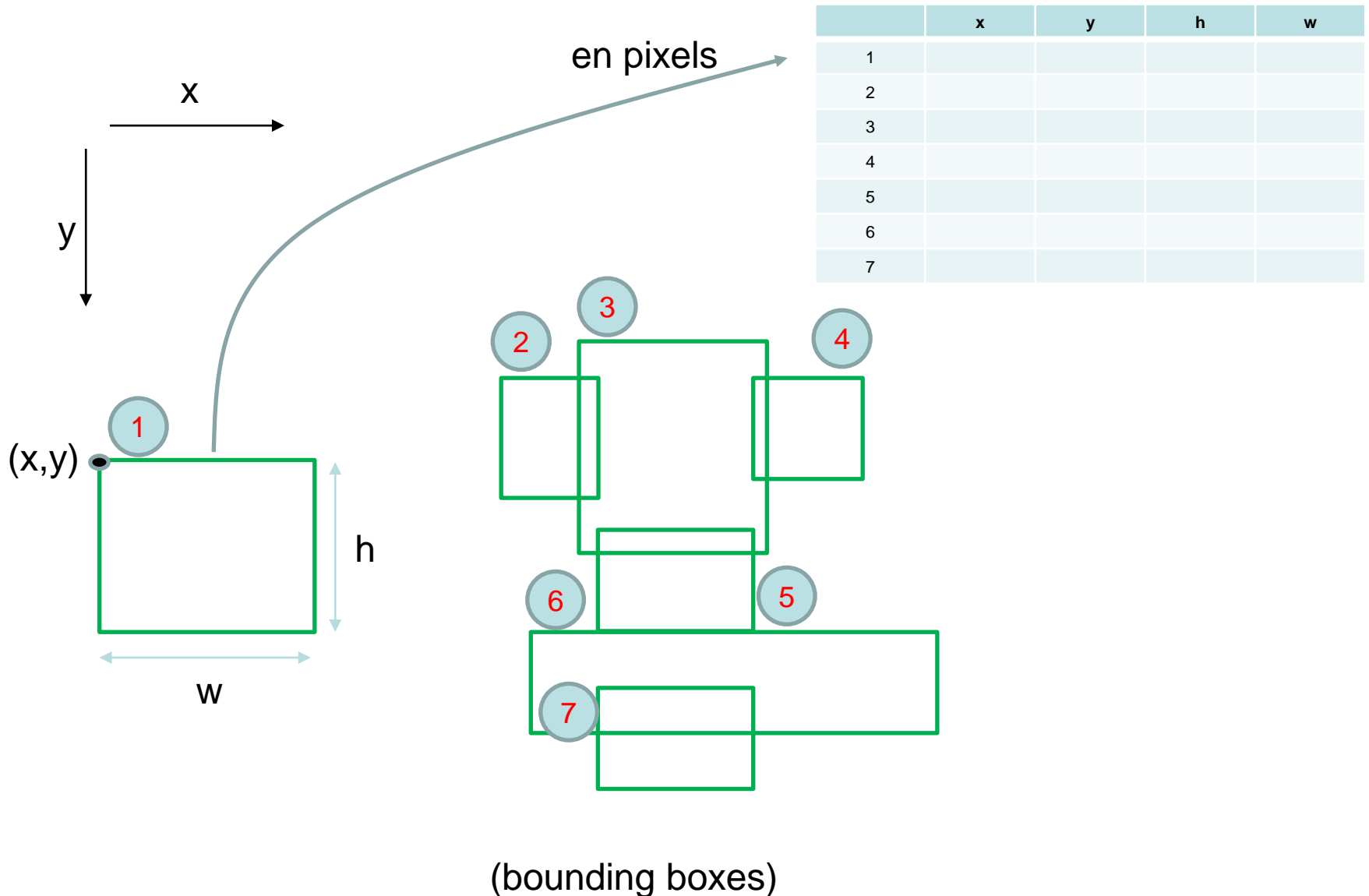
(bounding boxes)

Datos de Entrenamiento (2/2)



(bounding boxes)

Datos de Entrenamiento (2/2)



Formato

Las fotos deben ser a color en formato JPG con extension jpg (en minúsculas, no se acepta extensión jpeg, JPG, etc).

La foto debe ser tomada en modo horizontal (no vertical).

La resolución horizontal de la foto debe ser al menos 1600 pixels.

Formato

El nombre del archivo de la foto debe ser:

X_GG_NNNNN.jpg

X:

'R' para fotos con 100% rayado
'N' para fotos con 100% no rayado
'A' para fotos con algún rayado

GG:

Número de grupo
01, 02, etc.

NNNN:

Número consecutivo
de la foto
00001, 00002, etc.

Formato

El nombre del archivo de los bounding boxes debe ser (para la foto A_GG_NNNNN.jpg):

A_GG_NNNNN.txt

El archivo texto debe contener una fila por bounding box, cada fila contiene cuatro números enteros separados por una coma: x, y, h, w.

Entrenamiento (Training)

Para el entrenamiento se debe usar solo las imágenes y los bounding boxes de la carpeta 'Training'.

No está permitido el uso de Deep Learning.

Se recomienda usar patches (sub-ventanas pequeñas de dos clases, patches rayados y patches no rayados).

Pruebas (Testing)

Para las pruebas se debe usar solo las imágenes de la carpeta 'Testing'.

El output es una imagen binaria que indica un pixel como '1' si corresponde a una zona rayada y '0' para no rayada.

Pruebas (Testing)

A partir de los bounding boxes de Testing se genera una imagen binaria de clasificación ideal.

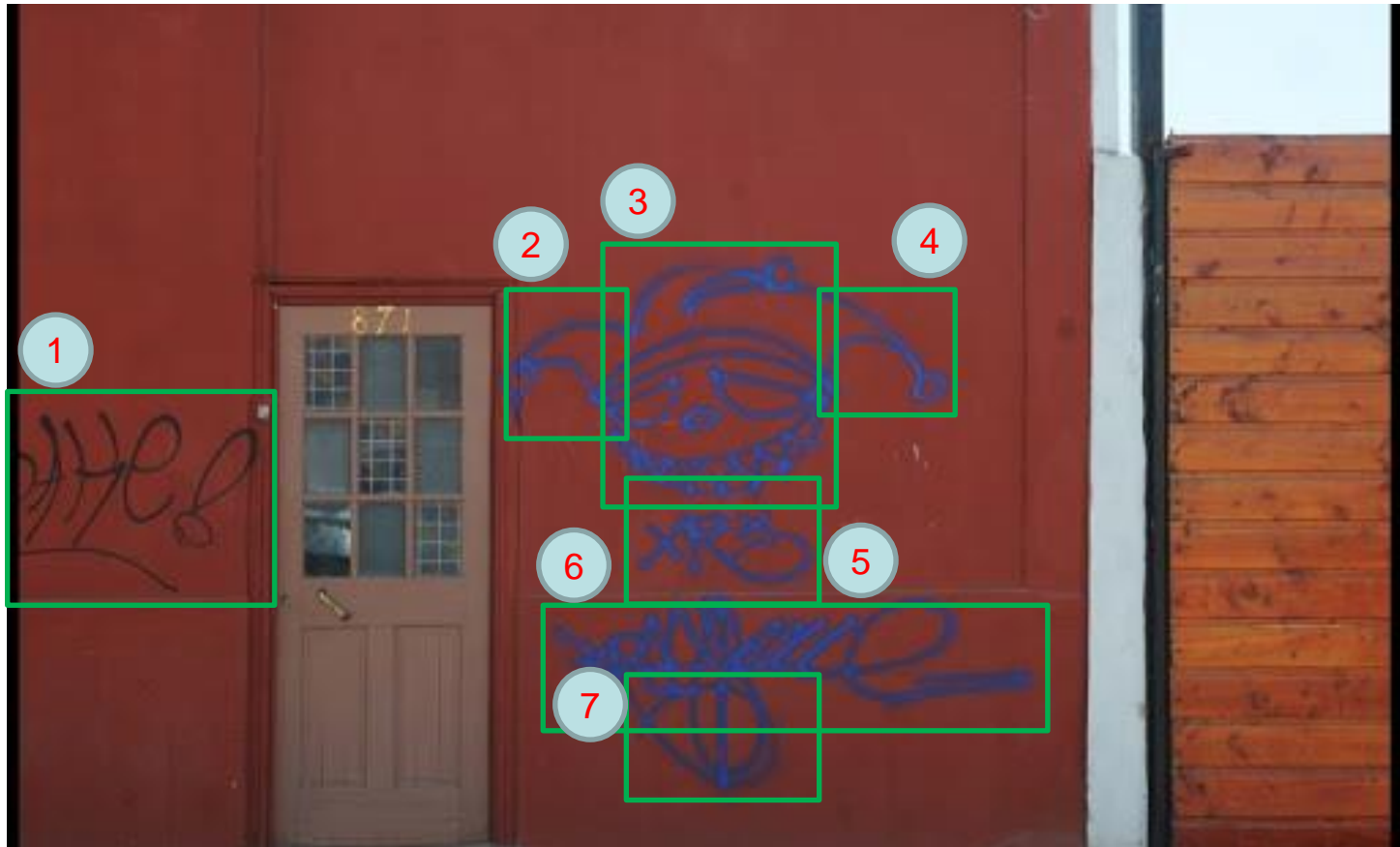
Se deberá calcular estadísticas de los pixels que estén bien y mal clasificados, y se entregará la métrica de 'precision' y 'recall'.

Ejemplo



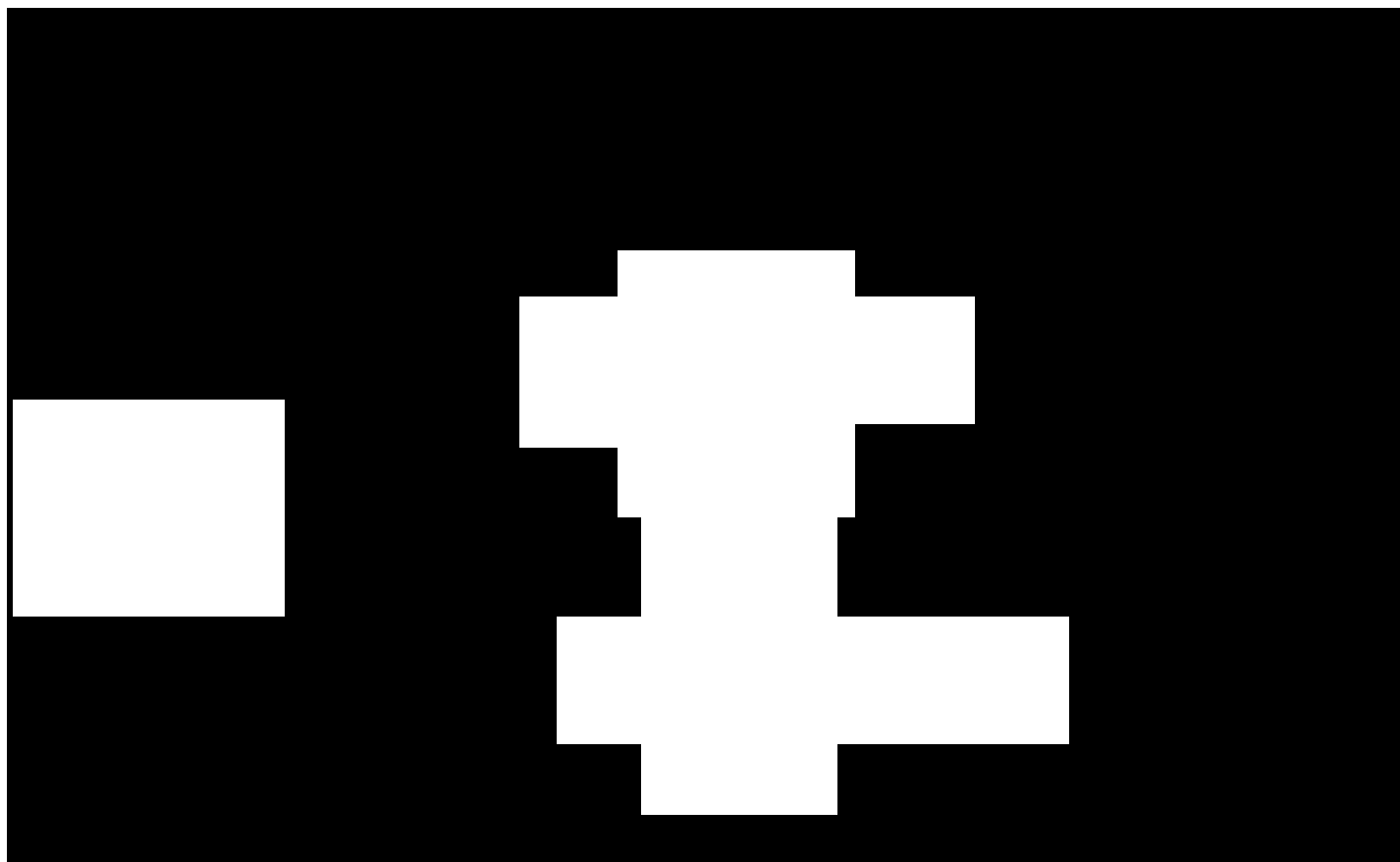
(imagen original)

Ejemplo



(bounding boxes)

Ejemplo



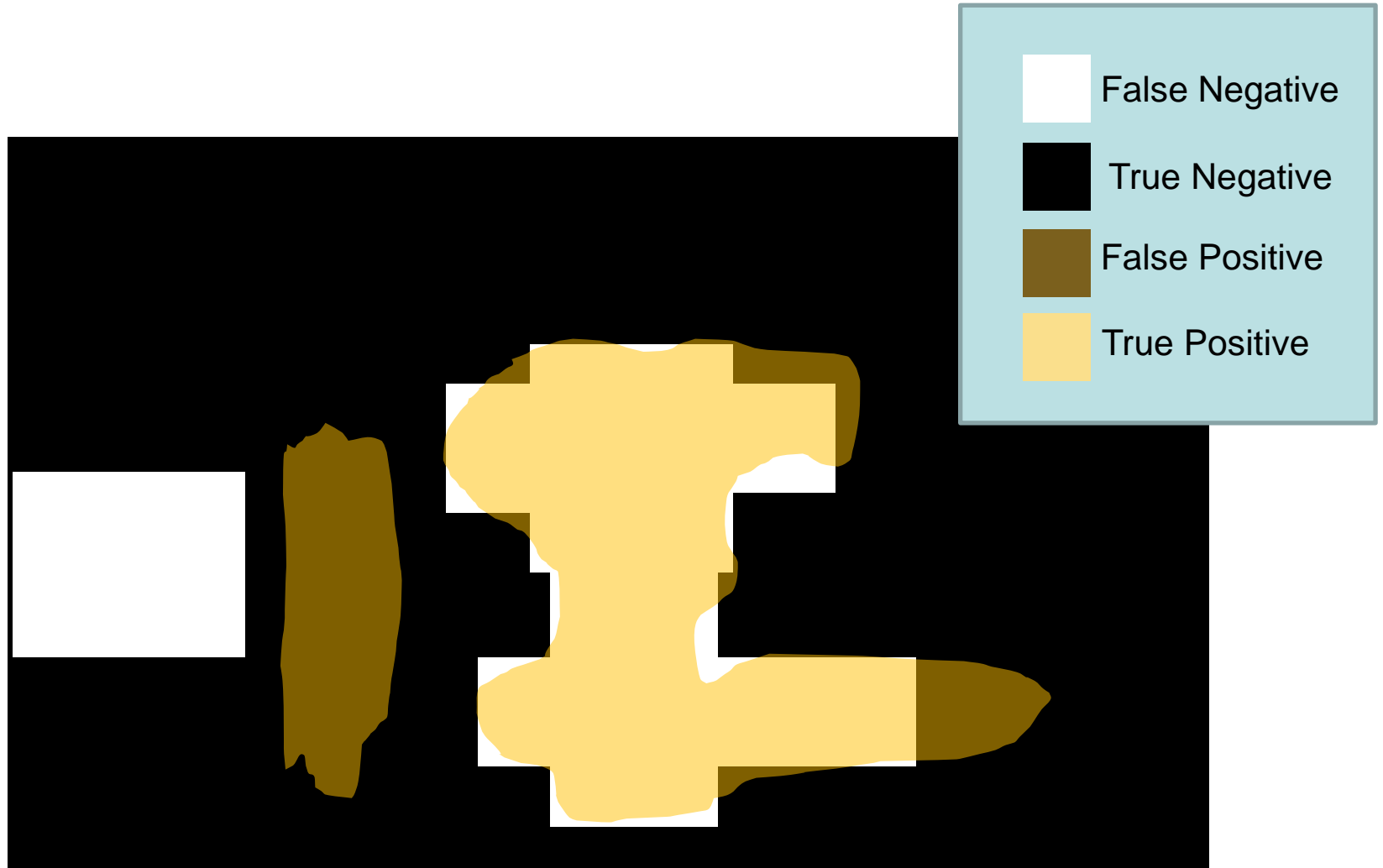
(detección ideal)

Ejemplo



(predicción: detección realizada por el algoritmo)

Ejemplo



(estadísticas)

Ejemplo

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

		Predicción →	
		1	0
Ideal ↓	1	True Positive	False Negative
	0	False Positive	True Negative

(estadísticas)